sentra.umm.ac.id

# Program Book

## "Industrial Revolution 4.0 : Sustainable Technology Development for Competitiveness and Responsiveness"

sentra
SEMINAR NASIONAL
TEKNOLOGI DAN REKAYASA

**Desember**
6 – 7, 2018

Aula GKB IV Lt. 9 Kampus 3
Universitas Muhammdiyah Malang

**Fakultas Teknik**
**Universitas Muhammadiyah Malang**

**2018**

UMM

ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050

# ANALISA PERFORMA APACHE HADOOP DENGAN H2O MENGGUNAKAN BENCHMARK HIBENCH VIA CLOUD COMPUTING

**Aminudin**
Teknik Informatika, Universitas Muhammadiyah Malang, Malang

Kontak Person:
Aminudin
Universitas Muhammadiyah Malang
E-mail: aminudin2008@umm.ac.id

***Abstrak***

*Perkembangan data yang kian cepat dan masif membuat media penyimpanan konvensional kesulitan untuk menyimpan, mengelola dan menganalisis data dengan ukuran yang besar. Untuk mengelola dan menganalisis data yang berukuran besar dapat menggunakan konsep High performance komputer (HPC), tetapi konsep HPC membutuhkan resource serta biaya yang mahal. Untuk mengatasi mahalnya biaya pada penggunaan HPC tersebut dapat menggunakan konsep paralel komputing. Paralel komputing adalah penggunaan beberapa komputer yang saling terhubung untuk mengolah data dalam ukuran yang besar. Salah satu tools paralel komputing yang lagi trend saat ini adalah Apache Hadoop. Apache hadoop dengan segala kelebihanya mampu memproses data yang besar secara parallel dengan membagi pemrosesan ke beberapa node yang digunakan. Berdasarkan dari pengujian yang dilakukan didapatkan Apache Hadoop mampu memproses data dengan algoritma Deep Learning dengan ukuran data sampai 6 GB yang dibagi menjadi empat node. Hasil pengujian berdasarkan parameter yang diujikan menggunakan Benchmark Hibench didapatkan nilai running time, throughput, average memory, average CPU dipengaruhi oleh ukuran data dan jumlah node yang digunakan.*

***Kata kunci:*** *Apache Hadoop, Benchmark Hibench, H2O*

## 1. Pendahuluan

Seiring perkembangan teknologi komunikasi dan informasi dan perkembangan aplikasi sosial media seperti facebook, twitter, instagram dsb, maka dapat dikatakan perkembangan data semakin hari semakin berkembang. Berdasarkan artikel yang di posting oleh DOMO tentang *Data Never Sleeps* pada 2015 menyatakan di setiap menitnya user facebook memposting 4,166,667 postingan, user instagram memposting sebanyak 1,736,111 photo, user Twitter mengirim 347,222 tweets dsb [1]. Data yang semakin besar ukurannya tersebut membuat data semakin sulit untuk dikoleksi, disimpan, dikelola maupun dianalisis dengan menggunakan sistem database biasa dikarenakan ukurannya yang terus bertambah, hal inilah yang disebut dengan konsep big data. International Data Corporation (IDC) memperkirakan ukuran data semesta digital berada pada angka 0.18 zettabytes (1 zettabytes = $1024^7$ bytes), serta meramalkan akan menjadi 10 kali lipatnya setiap 5 tahun [2]. Dari beberapa penjelasan diatas dibutuhkan suatu tools atau framework yang dapat mengolah dan memproses data dengan volume yang besar, serta menganalisis hasil yang didapat berdasarkan dari tools yang digunakan salah satunya adalah Apache Hadoop, Apache Spark, Apache Storm dan Apache Mahout[3].

Begitu pula dengan data cuaca, hampir setiap negara memiliki departemen meteorologi yang melakukan prediksi cuaca pada daerah tersebut. Salah satu data meteorologi dan klimatologi adalah NCDC (*National Centers for Environmental Information)*. Data tersebut secara gratis dapat diunduh untuk keperluan research terkait dengan banyak hal salah satunya penelitian tentang *big data*. NCDC merupakan salah satu database yang menyimpan arsip data iklim terbesar di dunia yang menyediakan layanan data klimatologis untuk seluruh pengguna di seluruh dunia. Data di dalam database NCDC tersebut terkumpul secara kolektif yang didapatkan dari sensor dengan frekuensi 3-4 kali per-jam[4]. Biasanya, data ini disimpan dalam format tidak terstruktur. Struktur dari format data ini adalah file teks biasa dimana masing-masing bidang dipisahkan oleh koma atau tab atau mungkin oleh titik koma. Jumlah data yang besar ini telah terakumulasi sejak bertahun-tahun terakhir dan akan terus bertambah. Pengolahan langsung data tidak terstruktur ini menggunakan metode dan alat konvensional sulit dan tidak efisien. Hal ini mengakibatkan tantangan penyimpanan dan pengolahan data cuaca yang sangat besar [5].

Untuk dapat menyimpan dan mengolah data yang berukuran besar (big data) secara baik dan cepat dibutuhkan teknologi komputer yang khusus yang disebut high performance computer atau super

ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050

komputer, akan tetapi untuk membangun suatu sistem super komputer membutuhkan biaya yang tidak murah. Untuk mengatasi masalah ini, maka platforms yang digunakan untuk menyimpan dan mengolah big data menggunakan sebuah sistem yang disebut parallel computing [6][7]. Parallel computing adalah penggunaan beberapa komputer yang saling terhubung untuk mengolah data dalam ukuran yang besar. Salah satu platform yang masih sering digunakan sampai saat ini untuk mengolah data yang berukuran besar (big data) secara terdistribusi dan dapat berjalan diatas cluster adalah Apache Hadoop dan Apache Spark.

Apache Hadoop adalah framework yang digunakan untuk pemrosesan data besar yang tersebar di seluruh cluster komputer dengan menggunakan data model pemrograman MapReduce[3]. Ada empat komponen hadoop, yaitu Map Reduce, Hadop Utilities, YARN (*Yet Another Resource Negotiator*) dan HDFS (*Hadoop Distributed File System*)[8]. Map Reduce menyediakan kerangka kerja pemrosesan paralel yang mungkin merupakan solusi yang lebih baik daripada multithreading. Multithreading membutuhkan pengetahuan dan keterampilan yang mahir bagi programmer dalam mengkoordinasikan setiap thread dan critical section dapat menyebabkan banyak masalah. Di dalam Map Reduce permasalahan seperti itu tereduksi, dikarenakan data dilewatkan secara eksplisit antara fungsi sebagai parameter atau nilai balik yang hanya dapat diubah oleh fungsi aktif pada saat itu[9]. Untuk melakukan pemrosesan data baik itu prediksi, klasifikasi dan klasterisasi Framework Hadoop membutuhkan tambahan library untuk melakukan pemrosesan machine learning tersebut, library itu adalah H2O[10].

H2O merupakan distributed machine learning yang open source dapat digunakan untuk sebagai pemrosesan machine learning. H2O mendukung pemrosesan machine learning seperti deep learning, Naive Bayes Classifier, k-means clustering dsb. Untuk meningkatkan kecepatan pemrosesan machine learning pada H2O maka dibutuhkan Hadoop agar pemrosesan machine learning bisa dilakukan dengan pemrosesan secara terdistribusi. Maka di dalam penelitian ini akan dilakukan pemrosesan data prediksi memanfaatkan library H2O dengan diproses secara terdistribusi menggunakan menggunakan Apache Hadoop dengan membagi beberapa node-node yang saling terhubung untuk memproses data secara parallel. Kemudian untuk menganalisis performa kecepatan Hadoop di dalam memproses data digunakan Banchmark Hibench untuk mengetahui parameter time eksekusi setelah dibebankan beberapa node di dalam pemrosesan, penggunaan memory dan CPU serta througput ketika memproses data[11].

## 2. Metode Penelitian

### 2.1 Dataset

Ada dua dataset yang digunakan di dalam penelitian ini, yaitu Storm Event Database (SED) dan Global Surface Summary of the Day (GSSOD). Data SED merupakan salah satu data yang dipublikasikan oleh National And Atmospheric Administration (NOAA). Saat ini, SED berisikan data yang dikumpulkan oleh *National Weather Service* per-Januari 1960 sampai Desember 2016. Karena perubahan dalam prosedur pengumpulan dan pengolahan yang terjadi setiap waktu, ada beberapa periode yang hanya mencatat fenomena cuaca dengan jenis tertentu. Selain itu, NOAA juga melakukan revisi dan standarisasi data tetapi tidak sampai mengubah nilai data dari Storm Event Database. Gambar 1 menunjukkan contoh data SED.

| EPISODE_ID | EVENT_ID | STATE | STATE_FIPS | YEAR | MONTH_NAME | EVENT_TYPE |
|---|---|---|---|---|---|---|
| 107480 | 644330 | MISSISSIPPI | 28 | 2016 | July | Thunderstorm Wind |
| 108769 | 651823 | SOUTH CAROLINA | 45 | 2016 | July | Heavy Rain |
| 108769 | 651825 | SOUTH CAROLINA | 45 | 2016 | July | Thunderstorm Wind |
| 108812 | 651828 | NORTH CAROLINA | 37 | 2016 | July | Thunderstorm Wind |
| 107252 | 643319 | MISSOURI | 29 | 2016 | July | Hail |
| 107253 | 643321 | ARKANSAS | 5 | 2016 | July | Thunderstorm Wind |
| 107254 | 643431 | TENNESSEE | 47 | 2016 | July | Thunderstorm Wind |
| 107254 | 643437 | TENNESSEE | 47 | 2016 | July | Hail |
| 107254 | 643440 | TENNESSEE | 47 | 2016 | July | Thunderstorm Wind |

**Gambar 1** Contoh Data Storm Event Database (SED)

Data yang kedua, yaitu data GSSOD dikumpulkan berdasarkan *Data The Integrated Surface Hour* yang diperoleh dari *U.S. Air Force Climatology Center*. GSSOD berisikan data mengenai

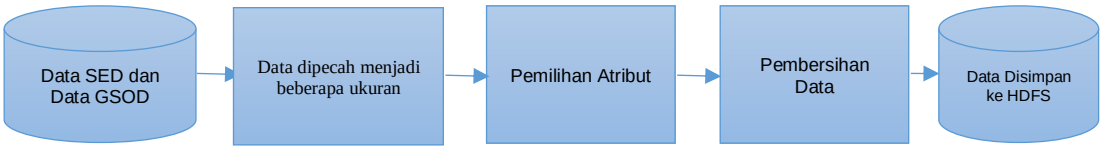
ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050

keadaan cuaca yang terpantau di sekitar area stasiun pengamatan seperti temperatur maksimum, temperatur minimum, kecepatan angin dll. Data tersebut mulai dikumpulkan per tahun 1929 hingga sekarang oleh sekitar 9000 stasiun pengamatan. Gambar 2 menampilkan contoh data dari GSSOD.

| STN--- WBAN | YEARMODA | TEMP | DEWP | SLP | STP | VISIB | WDSP |
|---|---|---|---|---|---|---|---|
| 010010 99999 | 20160101 | 35.0 24 | 31.7 24 | 997.1 24 | 995.9 24 | 6.5 | 6 |
| 010010 99999 | 20160102 | 36.3 24 | 32.9 24 | 1003.8 24 | 1002.5 24 | 5.7 | 6 |
| 010010 99999 | 20160103 | 33.0 24 | 29.5 24 | 1012.7 24 | 1011.5 24 | 5.8 | 6 |
| 010010 99999 | 20160104 | 35.0 24 | 33.2 24 | 1016.3 24 | 1015.1 24 | 1.7 | 6 |
| 010010 99999 | 20160105 | 34.6 24 | 29.7 24 | 1016.0 24 | 1014.8 24 | 9.3 | 6 |
| 010010 99999 | 20160106 | 35.0 24 | 29.0 24 | 1012.2 24 | 1011.0 24 | 11.4 | 6 |

**Gambar 2** Contoh Data Storm Event Database (GSSOD)

Implementasinya data dari Data SED akan dipecah menjadi 250 MB, 500 MB, 1 GB dan data dari GSSOD dipecah menjadi 2 GB, 4 GB dan 6 GB. Hal itu dilakukan agar pada pengujian dapat diketahui pola pemrosesan Hadoop ketika diberikan data yang bervariasi ukurannya (Gambar 3).




**Gambar 3** Proses pengolahan data

## 2.2 Kebutuhan Resource Sistem Hadoop di Cloud Computing

Pada penelitian ini diperlukan sebuah *computer cluster* dengan lima node yang disetting dengan satu *node* sebagai *cluster* dengan empat *node* lainya sebagai *slave*. Implementasi *computer cluster* menggunakan layanan cloud computing (Tabel 1), yaitu *Infrastructure as a Server* (IAAS) dengan nama BiznetGioCloud. Beberapa node yang dikonfigurasi tersebut memiliki spesifikasi perangkat keras yang sama tetapi memiliki perbedaan sedikit perbedaan ukuran media penyimpanan yaitu *master node* yang memiliki media penyimpanan yang lebih besar. Hal ini terjadi karena *master node* akan dipasang beberapa perangkat lunak inti, sehingga membutuhkan media penyimpanan yang besar. Media penyimpanan pada slave *node* lain akan menampung sementara beberapa dataset sebelum disalin ke dalam HDFS untuk mencegah kegagalan sistem disebabkan oleh ruang penyimpanan yang kurang. Hal tersebut dapat dimaklumi karena Apache Hadoop memerlukan ruang penyimpanan yang cukup untuk melancarkan komputasi terdistribusi.

Komputer *cluster* dipasang dengan sistem operasi CentOS untuk memudahkan proses komputasi maka di dalam *master node* dipasang Hortonwork Data Platform (HDP), sebuah *hadoop distributions* yang digunakan untuk memudahkan pemasangan *framework - framework big data* yang akan digunakan dalam penelitian ini. Pemasangan HDP juga dilakukan pengaturan SSH Public Key untuk membentuk *computer cluster* dengan menyambungkan empat *node* sehingga sangat mungkin untuk dilakukan komputasi terdistribusi. Selanjutnya, *master node* dipasang perangkat lunak lain untuk menunjang dalam pengujian sistem serta s*lave node* hanya menampung dataset - dataset yang akan digunakan dalam pengujian serta program tambahan untuk kemudahan pengguna dalam mengakses layanan perangkat lunak yang berjalan pada master *node*.

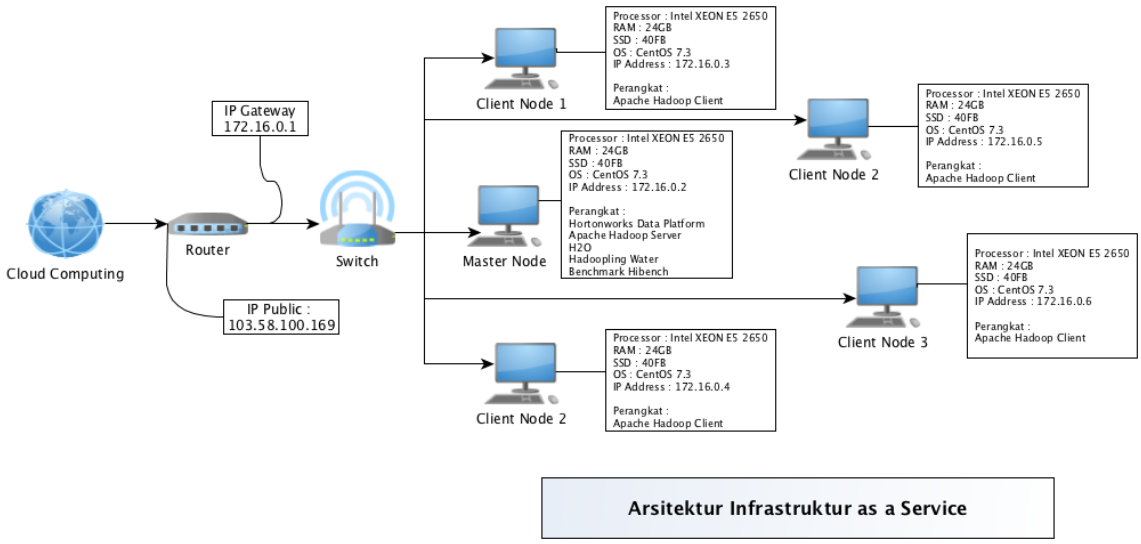
**Tabel 1** Kebutuhan perangkat keras pada *Cloud Computing*

| Nama | Spesifikasi Kebutuhan | | |
|---|---|---|---|
| | Processor | RAM | SSD |
| Master Node | Intel Xeon E5-2650 v3 (4) @ 2.299 GHz | 24 GB | 80 GB |
| Slave Node 1 | Intel Xeon E5-2650 v3 (4) @ 2.299 GHz | 24 GB | 40 GB |
| Slave Node 2 | Intel Xeon E5-2650 v3 (4) @ 2.299 GHz | 24 GB | 40 GB |
| Slave Node 3 | Intel Xeon E5-2650 v3 (4) @ 2.299 GHz | 24 GB | 40 GB |

ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050

## 2.3 Rancangan Topologi Sistem pada Cloud Computing

Berdasarkan kebutuhan sistem yang telah dilakukan, bahwa di dalam penelitian ini diperlukan sebuah *computer cluster* dengan jumlah lima *node*. Dari lima node tersebut satu *node* berperan sebagai *master node* sedangkan *node* lainnya berperan sebagai *slave node*. Rancangan *computer cluster* yang digunakan dalam penelitian ini ditunjukan pada Gambar 4. Pada rancangan tersebut *master node* dan *slave node* akan dipasang dengan perangkat lunak yang berbeda. Hal itu disebabkan karena adanya perbedaan dalam pembagian tugas *master node* dan *slave node*. Master *node* bertugas untuk mengatur seluruh *node* dalam *computer cluster* termasuk mempersiapkan sumber daya yang cukup agar dapat melakukan komputasi terdistribusi. Oleh karena itu, *master node* perlu dipasang perangkat lunak tertentu HDP agar hal tersebut dapat tercapai. *Slave node* bertugas untuk melakukan komputasi sesuai dengan yang diminta oleh *master node*. Jika *master node* tidak melakukan permintaan, maka *slave node* siap seperti sediakala. *Slave node* dipasang dengan program tambahan untuk mempermudah pengguna dalam mengakses layanan perangkat lunak yang berjalan pada *master node*. Sehingga pengguna dalam melakukan analisis data dimanapun tanpa perlu harus berada di *master node*. IP Address yang digunakan pada node yang digunakan menggunakan Network Address yang sama dengan IP Gateway 172.16.0.1. Selain itu, penggunaan IP Public 103.58.100.169 untuk mempermudah *computer cluster* dalam mengunduh data-data yang diperlukan melalui internet saat melakukan pemasangan perangkat lunak untuk masing-masing *node*. Seluruh perangkat keras yang terdapat pada Gambar 4 diimplementasikan dengan salah satu layanan cloud computing, yaitu IaaS.




**Gambar 4** Topologi rancangan sistem

## 2.4 Rancangan Pengujian

Setelah rancangan *computer cluster* diimplementasikan, maka berikutnya adalah pengujian sistem. Dalam penelitian ini, pengujian sistem akan dilakukan dengan menggunakan dua skenario. Skenario tersebut yaitu penggunaan jumlah *node* dan ukuran dataset yang berbeda oleh Apache Hadoop saat melakukan komputasi terdistribusi dengan menggunakan algoritma Deep Learning. Pengujian sistem dengan kedua skenario tersebut dilakukan menggunakan alur pengujian yang telah ditentukan agar menghasilkan data yang akurat dan valid untuk keperluan pengambilan kesimpulan. Selain itu, ada empat parameter pengujian yang digunakan sehingga dapat memberikan gambaran detail mengenai pengaruh jumlah *node* yang dipakai dan ukuran dataset yang diproses terhadap kinerja komputasi Apache Hadoop dengan menggunakan algoritma Deep Learning.

Skenario pertama akan dilakukan dengan cara menguji kinerja komputasi Apache Hadoop dalam memproses dataset menggunakan algoritma Deep Learning dengan penggunaan jumlah *node* tertentu dalam *computer cluster* dua sampai dengan empat node. Sedangkan skenario kedua dilakukan dengan cara mengukur kinerja komputasi Apache Hadoop dalam memproses ukuran dataset yang berbeda dengan rincian pengujian dengan ukuran 250 MB, 500 MB dan 1 GB akan menggunakan SED, sedangkan pengujian dengan ukuran 2 GB, 4 GB dan 6 GB akan menggunakan GSSOD.

ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050

Pengujian sistem dalam penelitian ini menggunakan empat parameter pengujian yang telah ditentukan. Nilai parameter pengujian tersebut didapatkan melalui Benchmark Hibench. Benchmark Hibench merekam seluruh aktivitas komputer atau sebuah *computer cluster* saat sebuah proses sistem atau perangkat lunak dijalankan mulai dari awal waktu saat dieksekusi hingga akhir waktu saat berhenti. Parameter itu antara lain *running time, throughput, average memory usage, average CPU usage,*

## 3. Hasil dan Pembahasan

### 3.1 Implementasi Sistem

Rancangan tersebut selanjutnya diimplementasikan dalam sebuah layanan *Infrastructure as a Service* (IAAS), yaitu BiznetGioCloud. Untuk implementasi perangkat keras dan perangkat lunak, BiznetGioCloud telah menyediakan hal tersebut sehingga penulis hanya cukup menggunakan saja tanpa perlu melakukan konfigurasi yang rumit.

### 3.2 Pembersihan Data

Data Storm Event Database (SED) dan Global Surface Summary of The Day (GSSOD) selesai diunduh, maka kedua dataset tersebut dilakukan pembersihan. Pembersihan dilakukan dengan cara restrukturisasi kolom yaitu menambahkan kolom baru berdasarkan nilai dari kolom yang ada atau menghapus kolom yang tidak diperlukan. Biasanya ini terjadi pada kolom tanggal yang memiliki format tertentu sehingga perlu diekstrak menjadi kolom - kolom baru sesuai dengan kebutuhan. Selanjutnya, kolom tanggal tersebut dapat dihapus untuk meminimalkan ukuran dataset.

Setiap kolom tidak selalu memiliki data, dalam hal ini penulis tetap menggunakan data tersebut untuk menjaga orisinalitas dalam pengujian. Setelah pembersihan data dirasa cukup, kedua dataset tersebut disimpan ke dalam Hadoop Distributed File System dengan ukuran yang berbeda-beda untuk keperluan pengujian. Storm Event Database akan disimpan dengan ukuran masing-masing 250 MB, 500 MB dan 1 GB. Sedangkan Global Surface Summary of The Day akan disimpan dengan ukuran masing-masing 2 GB, 4 GB dan 6 GB.

### 3.3 Mengumpulkan Hasil Pengujian

Pengujian sistem dilakukan dengan menggunakan program Benchmark Hibench. Benchmark Hibench adalah program yang dapat merekam aktivitas komputer atau sebuah *computer cluster* saat sebuah proses sistem atau perangkat lunak dijalankan, mulai dari awal dieksekusi hingga waktunya berhenti. Dalam pengujian ini, proses sistem atau perangkat lunak yang dijalankan tersebut adalah sebuah beban kerja. Banyaknya beban kerja yang perlu dijalankan dalam pengujian sistem, disesuaikan dengan banyaknya variabel penelitian yang telah ditentukan sebelumnya.

Benchmark Hibench mulai bekerja saat sebuah beban kerja dijalankan. Setelah beban kerja tersebut berhenti, maka Benchmark Hibench secara otomatis akan mengumpulkan hasil rekaman aktivitas sistem mulai dari awal waktu hingga akhir waktu dan menyimpannya dalam bentuk teks. Teks tersebut kemudian diurai untuk mendapatkan nilai - nilai parameter pengujian yang selanjutnya dikumpulkan untuk keperluan analisis.

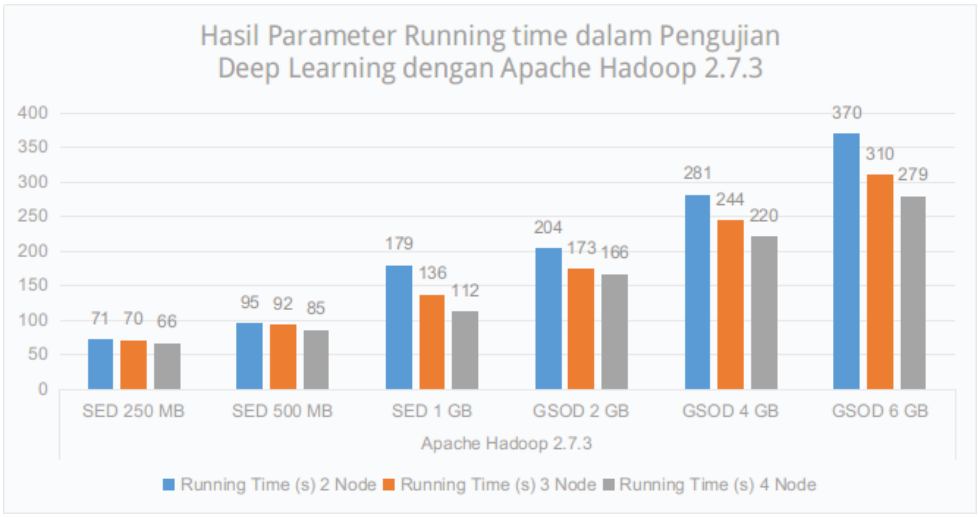
### 3.3 Hasil Pengujian

#### a. Running Time

*Running Time* adalah waktu yang diperlukan oleh Hadoop dalam memproses data. Nilai parameter ini didapatkan dengan cara mencari selisih antara waktu awal dan waktu akhir saat Apache Hadoop dijalankan atau dihentikan untuk memproses dataset dalam *computer cluster* dengan menggunakan algoritma Deep Learning [11]. Hasil pengujian Hadoop pada parameter Running time ditunjukkan pada Gambar 5.

Berdasarkan hasil dari pengujian *running time* bahwasanya Apache Hadoop di dalam memproses data SED maupun GSSOD bersifat linear, artinya ukuran data dan jumlah node yang digunakan mempengaruhi waktu running time. Hal ini sesuai dengan konsep dari Hadoop di mana lama waktu sebuah pemrosesan di dalam framework big data sangat dipengaruhi dari ukuran data dan jumlah node yang digunakan untuk pemrosesan data. Intinya semakin besar data yang digunakan semakin lama waktu pemrosesan dan semakin banyak node yang dipakai maka semakin cepat running timenya.

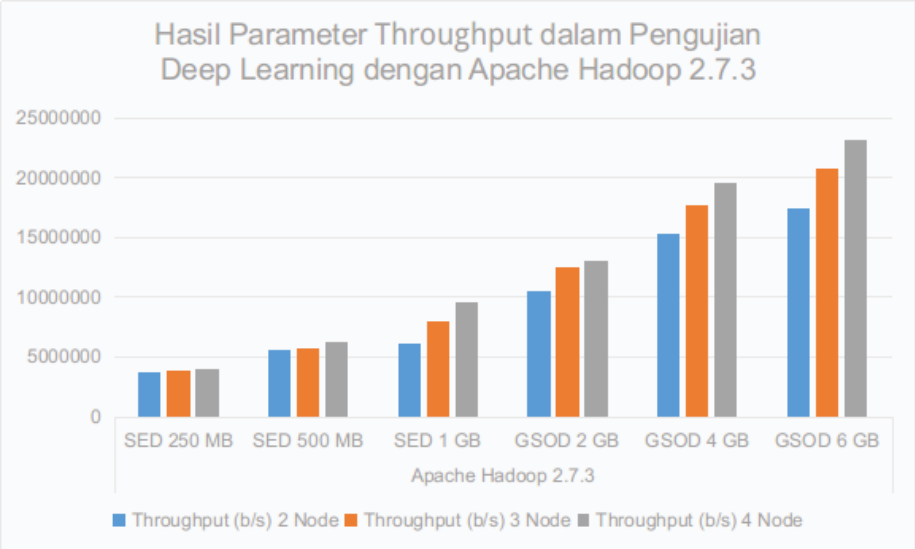
ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050




**Gambar 5** Hasil pengujian Running Time pada Hadoop

## b. Throughput

*Throughput* adalah kecepatan dalam bertukar data dengan ukuran tertentu[11]. Kegiatan bertukar data tersebut terjadi pada *node - node* yang dipakai dalam *computer cluster*, saat Apache Hadoop memproses dataset. Oleh karena itu, semakin tinggi nilai *throughput* yang dicapai, maka semakin sedikit waktu yang dibutuhkan Apache Hadoop untuk menyelesaikan komputasi. Hasil pengujian Hadoop yang ditampilkan pada Hibench ditunjukkan pada Gambar 6.



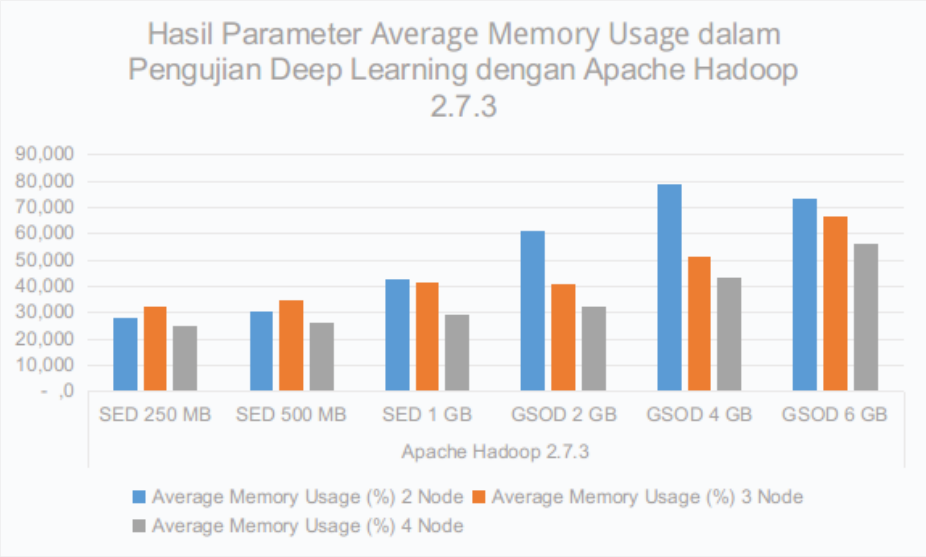

**Gambar 6** Hasil pengujian Throughput pada Hadoop

Apache Hadoop memiliki pertukaran data yang cepat, hal ini dapat dibuktikan dengan hanya memproses data 250 MB Hadoop mampu bertukar data 4 jutaan bit/sec, dan pertukaran data tersebut akan naik manakala jumlah node ditambah. Maka dari itu hasil pengujan throughput ini menunjukan bahwa memang Apache Hadoop sangat cocok digunakan untuk melakukan pemrosesan dengan jumlah data yang besar.

## c. Average Memory Usage

*Average memory usage* adalah persentase rata-rata penggunaan *memory* pada *node - node* yang digunakan saat Apache Hadoop saat memproses dataset. Nilai parameter ini didapatkan dari berkas hasil rekaman aktivitas sistem yang dihasilkan oleh Benchmark Hibench. Satuan pengukuran yang dipakai adalah persentase (%). Berikut Gambar 7 menunjukkan hasil untuk pengujian sistem dengan parameter *average memory usage*.

ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050

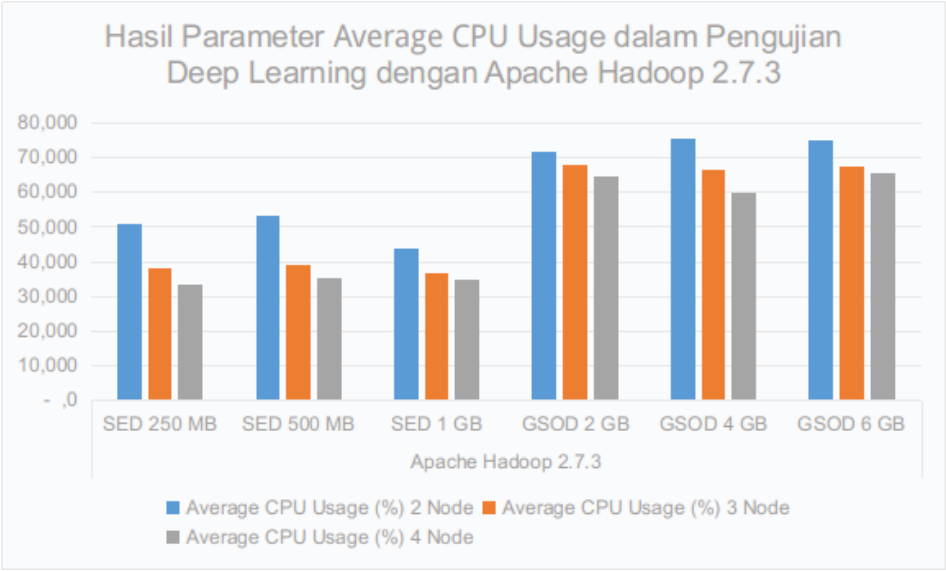



**Gambar 7** Hasil pengujian Throughput pada Hadoop

Gambar 7 menunjukkan bahwa Apache Hadoop pemrosesan data yang masih terbilang kecil Hadoop membutuhkan konsumsi memori yang konstan artinya masih dipengaruhi oleh ukuran data dan banyaknya node. Tetapi ketika Hadoop memproses data dengan ukuran yang besar misal 4 dan 6 GB perubahan konsumsi memory terlihat bahwa konsumsi memory yang dibutuhkan Hadoop lebih banyak memproses data 4 daripada 6 GB. Hal ini diakibatkan banyak faktor diantaranya adalah susunan struktur dataset yang beragam, lalu lama proses pengiriman data dari HDFS ke Map-Reduce, atapun software tambahan yang digunakan untuk mengintegrasikan antara Hadoop dan H2O yaitu Hadoopling water juga turut mempengaruhi konsumsi memory. Hal ini dapat dimaklumi karena pemrosesan dari Hadoop masih bersifat Batch atau antrean belum bersifat streaming seperti yang dimiliki oleh Apache Spark, Flink, Kafka dll.

### d. Average CPU Usage

*Average CPU usage* adalah persentase rata - rata penggunaan CPU pada *node - node* yang digunakan saat Apache Hadoop memproses dataset. Nilai parameter ini didapatkan dari berkas hasil rekaman aktivitas sistem yang dihasilkan oleh Hibench. Satuan pengukuran yang dipakai adalah persentase (%). Berikut Gambar 8 yang digunakan untuk hasil pengujian sistem dengan parameter *average CPU usage*.




**Gambar 8** Hasil pengujian average CPU Usage pada Hadoop

ISSN (Cetak) 2527-6042
eISSN (Online) 2527-6050

Nilai *average CPU usage* dalam pengujian Apache Hadoop menunjukkan bahwa kinerja rata-rata CPU masih dipengaruhi oleh jumlah node dan ukuran data. Gambar 8 menunjukkan bahwa semakin banyak node yang digunakan untuk memproses data penggunaan CPU cenderung menurun.

## 4. Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan dapat ditarik beberapa kesimpulan diantaranya:

1. Framework Hadoop dapat digunakan untuk memproses data dengan jumlah besar hal tersebut terlihat dari hasil pengujian yang ditunjukan pada parameter running time dan hasil throughput.
2. Proses komputasi Apache Hadoop bersifat linear yang artinya semakin besar ukuran data maka waktu yang diperlukan semakin lama dan juga semakin banyak node yang digunakan di dalam komputasi waktu yang diperlukan semakin sedikit.
3. Kebutuhan *average memory usage* di dalam Hadoop dipengaruhi oleh susunan struktur data, proses pengiriman data dari HDFS ke Map-Reduce, dan penggunaan *tools* integrasi antara Hadoop dengan H2O, yaitu Hadoopling Water sehingga konsumsi memory dari hasil pengujian lebih besar pemrosesan data dengan 4 daripada 6 GB.

Setelah melakukan penelitian ini, ada beberapa saran untuk mengembangkan penelitian ke depan, diantaranya:

1. Munculnya *framework* baru seperti Apache Flink, Kafka, Splunk yang diklaim lebih cepat dari Apache Hadoop dapat dijadikan bahan pengujian untuk melakukan penelitian kinerja komputasi lebih lanjut.
2. Saat ini, *library* algoritma Deep Learning berlomba - lomba agar dapat digunakan dengan GPU. Tentu saja sangat diharapkan jika ada penelitian selanjutnya yang membahas tentang kinerja komputasi algoritma Deep Learning dengan menggunakan GPU.
3. Selain H2O, banyak *library* algoritma Deep Learning seperti TensorFlow, Caffe, MXNET, DeepLearning4J, dll. Adanya fakta ini tentu akan membuka peluang untuk penelitian dalam menguji kinerja komputasi dari masing-masing *library.*

## Referensi

[1] L. Liu, “Performance comparison by running benchmarks on hadoop, spark, and hamr,” 2015.
[2] I. Mavridis and H. Karatza, “Performance evaluation of cloud-based log file analysis with Apache Hadoop and Apache Spark,” *J. Syst. Softw.*, vol. 125, pp. 133–151, 2017.
[3] S. Pan, “The Performance Comparison of Hadoop and Spark,” 2016.
[4] A. Zaslavsky, C. Perera, and D. Georgakopoulos, “Sensing as a Service and Big Data,” *Proc. Int. Conf. Adv. Cloud Comput.*, pp. 21–29, 2012.
[5] C. Engineering, P. Chouksey, and A. S. Chauhan, “Weather Data Analytics using MapReduce and Spark,” vol. 6, no. 2, pp. 42–47, 2017.
[6] A. Aminudin and M. Alwi, “Analisa Multithreading Pada Sistem Rekomendasi Menggunakan Metode Collaborative Filtering Dengan,” *Techno. Com*, vol. 17, no. 1, pp. 1–11, 2018.
[7] P. Khusumanegara, “Analisis Performa Kecepatan Mapreduce Pada Hadoop Menggunakan Tcp Packet Flow,” p. 72, 2014.
[8] S. Humbetov, “Data-intensive computing with map-reduce and Hadoop,” *2012 6th Int. Conf. Appl. Inf. Commun. Technol. AICT 2012 - Proc.*, 2012.
[9] J. Woo, “Apriori-Map / Reduce Algorithm.”
[10] S. S. Y. Ng, W. Zhu, W. W. S. Tang, L. C. H. Wan, and A. Y. W. Wat, “An independent study of two deep learning platforms - H2O and SINGA,” *2016 IEEE Int. Conf. Ind. Eng. Eng. Manag.*, pp. 1279–1283, 2016.
[11] Y. Samadi, M. Zbakh, and C. Tadonki, “Comparative study between Hadoop and Spark based on Hibench benchmarks,” 2016.